NFC smartcards in Python

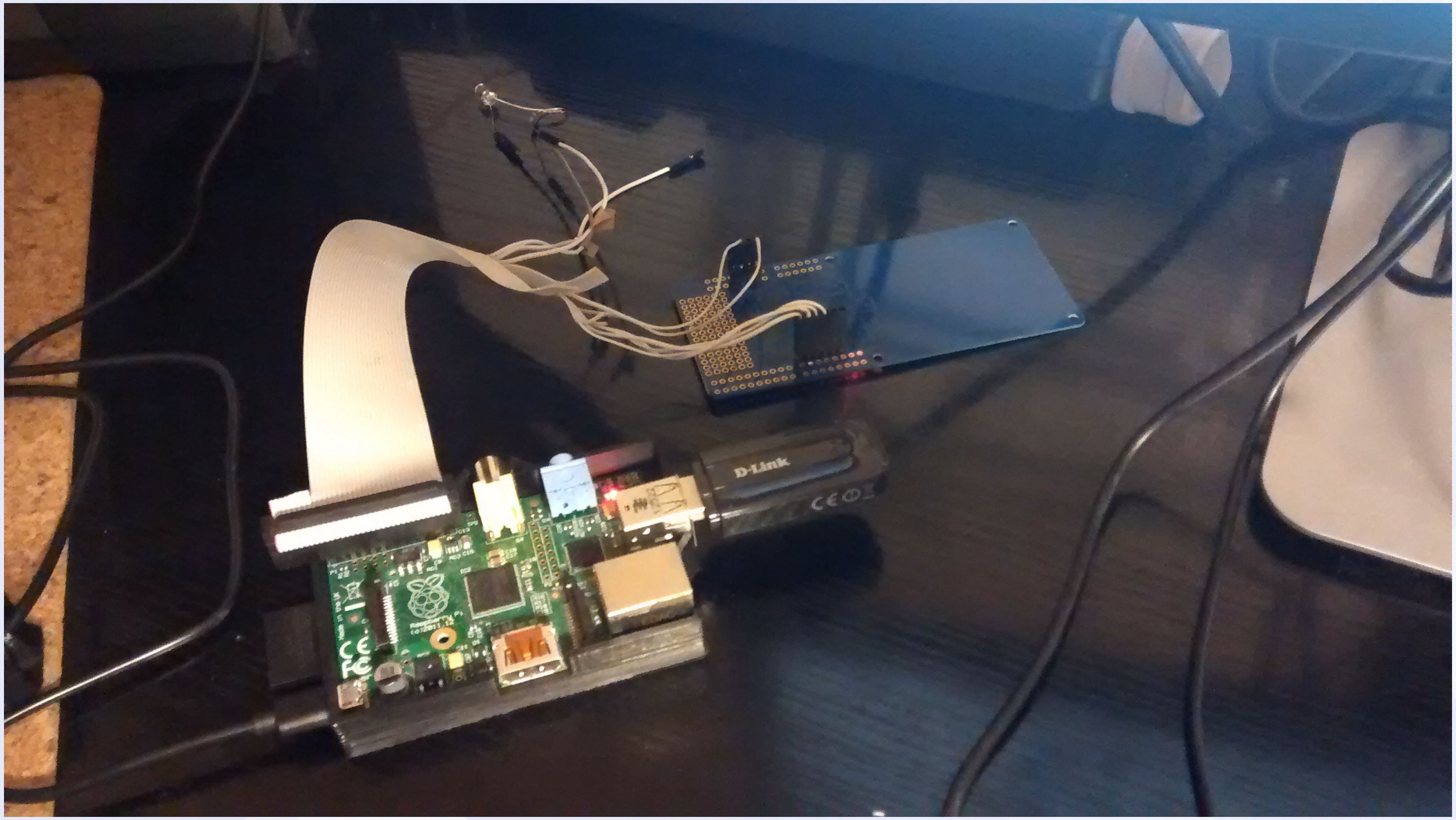
wrapping libnfc for use with smartcards

Ondrej Mikle • ondrej.mikle@gmail.com • 16.12.2015

First API for APDU over NFC

- APDUs are „assembler for smartcards“
- only API for NFC smartcards in python
- all other projects aim at simpler cards
  - Mifare Classic usually
- smartcards are much more interesting
  - Desfire
  - Yubikey Neo
  - EMV (Visa, Mastercard)

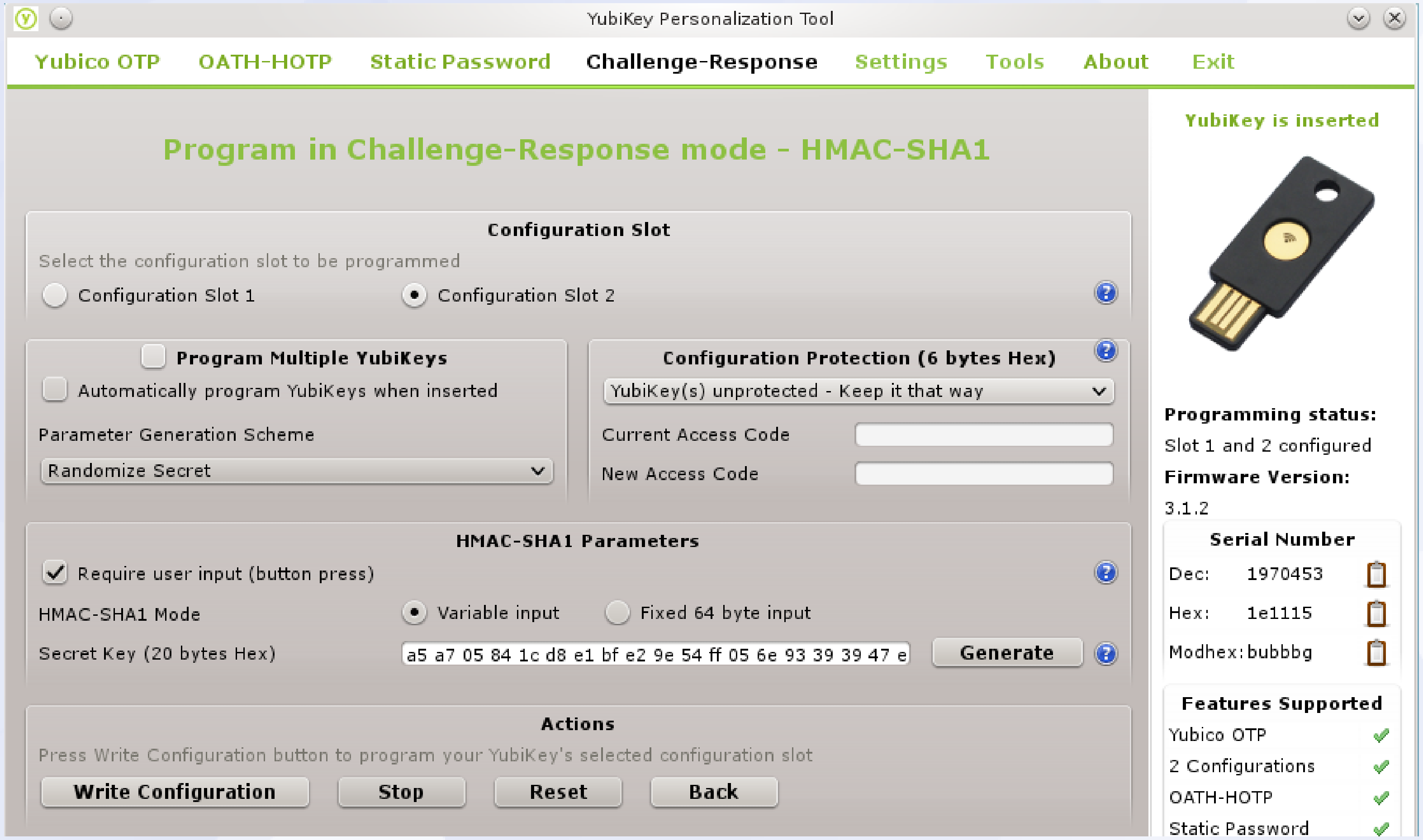
Raspi with PN532 over SPI



Demo app - authenthicator

- waits for card to be in the reader's field
- reads UID
  - looks at DB how this UID should authenthicate
  - either just UID or Yubikey's HMAC-SHA1
- if successful, wiringPi triggers a pin (lock)

Yubikey programming



YubiKey Personalization Tool

Program in Challenge-Response mode - HMAC-SHA1

Configuration Slot

Select the configuration slot to be programmed

Configuration Slot 1 [✓ Configuration Slot 2]

Program Multiple YubiKeys ☐

☐ Automatically program YubiKeys when inserted

Parameter Generation Scheme: Randomize Secret

Configuration Protection (6 bytes Hex)

YubiKey(s) unprotected - Keep it that way

Current Access Code:

New Access Code:

HMAC-SHA1 Parameters

✓ Require user input (button press)

HMAC-SHA1 Mode: [✓ Variable input] Fixed 64 byte input

Secret Key (20 bytes Hex): a5 a7 05 84 1c d8 e1 bf e2 9e 54 ff 05 6e 93 39 39 47 e

Actions

Press Write Configuration button to program your YubiKey's selected configuration slot

YubiKey is inserted

Programming status:
Slot 1 and 2 configured
Firmware Version:
3.1.2

Serial Number

Dec:	1970453
Hex:	1e1115
Modhex:	bubbbg

Features Supported

Yubico OTP	✓
2 Configurations	✓
OATH-HOTP	✓
Static Password	✓

Demo app - log

```
pi@raspberry1 ~/brmdoor_libnfc sudo python brmdoor_nfc_daemon.py
brmdoor_nfc.config

2015-12-04 17:05:16,305 INFO Unknown UID 80798c69
[brmdoor_nfc_daemon.py:128]
2015-12-04 17:05:23,782 INFO Unknown UID 80f02118
[brmdoor_nfc_daemon.py:128]
2015-12-04 17:05:29,130 INFO Unlocking for UID (uid: 22623733, nick:
UidMifare2) [brmdoor_nfc_daemon.py:116]
2015-12-04 17:05:38,711 INFO Unknown UID 805539bc
[brmdoor_nfc_daemon.py:128]
2015-12-04 17:05:45,117 INFO Unlocking after HMAC for UID (uid:
04372ED2A52E80, nick: YubikeyOld) [brmdoor_nfc_daemon.py:124]
```

Other demos

- see test_nfc.py
- reading NFC NDEF message
- HMAC-SHA1 on the Yubikey
- Visa read Track 2 Equivalent Data
- Mastercard execute and sign payment

How it's implemented

- „classic swig“ wrap of libnfc
  - in C++ because we want exception handling to propagate into Python
- APDUs were actually real pain to get working as there was minimal documentation
- nfc_smartcard.cpp has sending, receiving, parsing APDU

Use in Python

```
from binascii import hexlify
from nfc_smartcard import NFCDevice, NFCError

hex_apdus = [ # this asks for NDEF message stored on card (Yubikey/Desfire)
    "00 A4 04 00 07 D2760000850101",    #select NDEF application
    "00 a4 00 0c 02 E104",              # select NDEF message file 0xE104
    "00 b0 00 00 30" ]                  # read up to 0x30 bytes from record

# turn APDUs to binary
apdus = [hex_apdu.replace(" ","").decode("hex") for hex_apdu in hex_apdus]

nfc = NFCDevice()
uid = nfc.scanUID()
try:
    for apdu in apdus:
        rapdu = nfc.sendAPDU(apdu)
        print "Response SW %04x, data %s" % (rapdu.sw(), hexlify(rapdu.data()))
except NFCError, e:
    print "Failed to transmit APDU:", e.what()

nfc.close()
nfc.unload()
```

Project link

- https://github.com/hiviah/brmdoor_libnfc

Thanks

Ondrej Mikle • ondrej.mikle@gmail.com